# Data Wrangling

## What is data wrangling?

It is the process of cleaning, organizing & transforming raw data into usable format for analysis.

- creating new columns
- handling large amount of messy data

Need to load
→ Tidyverse
→ Install packages
→ 8 dif packages

## Select

→ Select column (mostly use)
: → between columns to select all column (in order)

code:

Select (') function

Rename <- select (data, column1, clum2 : column5, Column 6)

Filter function [ filter( ) ]

→ Subsetting / filtering data

- subset data fairly easy

Example

filtering Treatment "conv" from data

head (filter ( data , Treatment == "conv" ))

↓

Head will display first 6 lines

complex using & or !

head (filter ( data , Treatment == 'conv" &
& fungicide == "c" ))

↓

Display first 6 lines

* function mutate ( )

- To create new column

Code: converting tricliness to log transform

Base R

data$ Newcolumn <- log (data $tricliness
or var)

Advanced

head (mutate (data, Newcolumn = log (var) ))

## Pipe

→ Combining multiple functions together

→ combine output from one function into input of another function

→ Input the previous data from the right side of the pipe into the function on the left %>%

example:

combining whatever we've done so far

(mutate, filter, select)

facing right

data %>% select(

- input this data frame to this function
- don't need to supply data frame in each function using pipes

final code

what it's doing

selecting ← data %>% select( column1, column2, column3: column5, column6) %>%

subsetting ← filter( column4 == "conv") %>%

creating new column ← mutate (Newcolumn = log(Oldcolumn)) %>%

head ( )

## Summarise data

→ To find sd, mean etc.

code:

```
data %>% select ( ) %>%
 filter ( ) %>%
 mutate ( ) %>%
 summarize (outputname = mean (Newcolumn))
```

Output

| Mean.rich | ← output name |
|---|---|
| 1 | ← mean of the new column |

# connecting multiple summary statistics

```
previous pipeline %>%
summarize (Mean.rich = mean (New column),
                       ↓ output nm
           n = n( ),
           Sd. dev = sd (NewColumn)) %>%
```

To create a new column of sd.error. ← `mutate (std.error = sd.dev / sqrt(n))`

Output

| Mean.rich | n | sd.dev | std.error |
|---|---|---|---|
| | X | y | z |

function: group_by()

Summarize data by group

⇒ Calculating summary statistics by group

Code:

```
data %>% %
select( ) %>%
group_by( Var1, Var2) %>%
mutate( Newcol = log(Old col)) %>%
summarise( ) ;
  n = n( )
  sd.dev = sd (Newcol))
  mutate (Std.error = sd.dev/sqrt(n))
```

Connecting to plotting

→ output of plot and connect to ggplot

```
Previous code %>%
ggplot (aes(x = Var1, y = Var2))

geom_bar( stat = "Var1")
geom_errorbar (aes (x = Var1, ymin = Var2, St.error
                    ymax = Var2 + Stderror), width=0.4)
+
theme_minimal ( )
x Lab(" ")
y Lab( " Label" ) +
facet_wrap (~Var4)
```

Joining function

→ left_join( )
Keep all rows of X, add matching from Y

→ right_join( )

→ right_join → all Y, match X
(Reverse of left join)

→ inner_join
only keep rows common to both X and Y

→ full_join
→ keep all from X & Y

→ combining various set of data (?)
based on similar column

code:

```
# selecting just richness and sample ID

richness <- data %>%
  select(sample ID, richness)

# selecting column that don't include richness
metadata <- data %> %
  select(Sample ID, other columns)
# Adding the richness data to the metadata based on
# the common column of sample ID
head(left_join(metadata, richness, by = "SampleID"))
```

Pivot function

→ converting from wide to long data (one observation per sample)

i) -pivot_wider

convert long format to wide

ii) pivot_longer

wide → long

What is wide or long data

Long data

- Each row rep. a single observation
- values stack in one column
- for statistical analysis

Wide data

- Each row represents single subject or experimental unit
- Diff categories are stored in separate column.